Performance Evaluation of Anomaly-Detection Algorithm for Keystroke-Typing based Insider Detection

Liang He
National Key Laboratory of Science and Technology on Blind Signal Processing
P. O. Box 666
Chengdu, Sichuan, China 610041
lianghe@sei.xjtu.edu.cn

Zhixiang Li
National Key Laboratory of Science and Technology on Blind Signal Processing
P. O. Box 666
Chengdu, Sichuan, China 610041
lizx07@org.tsinghua.cn

Chao Shen
MOE KLINNS Lab,
Xi'an Jiaotong University
P. O. Box 1088
Xi'an, Shaanxi, China 710049
chaoshen@mail.xjtu.edu.cn

ABSTRACT

Keystroke dynamics is the process to identify or authenticate individuals based on the typing rhythm behaviors. There are many classifications proposed to check the user's legitimacy, and therefore we should make it clear how they perform in order to confirm promising research direction. Nevertheless, these researches provide experiments in different situations such as datasets, conditions and methodologies as well. This paper aims to benchmark the algorithms in the same dataset and feature in order to measure the performance on an equal level. Using dataset containing 51 subjects' typing rhythm, we implemented and evaluated 13 classifiers measured by F1-measure. We also develop a way to process the typing data, and test it on these algorithms. Considering the case that the model should reject outlander, we test the algorithms on open set. The top-performing classifier achieves F1-measure rates 0.92 when using 50 subjects' typing normalized data to train and the remaining one to test. The results, along with the normalization methodology, constitute a benchmark for comparing classifiers and measuring performance of keystroke dynamics for insider detection.

CCS CONCEPTS

• **Security and privacy → Intrusion detection systems**; *Formal security models*; *Access control*; *Operating systems security*;

KEYWORDS

keystroke dynamics; insider identification; F1-measure; normalization.

ACM Reference format:
Liang He, Zhixiang Li, and Chao Shen. 2017. Performance Evaluation of Anomaly-Detection Algorithm for Keystroke-Typing based Insider Detection. In *Proceedings of ACM TUR-C '17, Shanghai, China, May 12-14, 2017,* 8 pages.
DOI: http://dx.doi.org/10.1145/3063955.3063987

ACM TUR-C '17, Shanghai, China
 ISBN 978-1-4503-4873-7/17/05…$15.00
DOI: http://dx.doi.org/10.1145/3063955.3063987

1 INTRODUCTION

Insider attack has always been a major threat for information and cyber security of enterprise and government agencies. The insiders are very difficult to detect and defend against, and always exploit the privileges that do not belong to him for malicious purpose, such as stealing confidential data or obtaining root privileges on a system. The threats from these insider attackers have overtaken malwares as the most reported security incident according to recent reports from US National Threat Assessment Center [20] and Australian Cyber Security Centre [5, 6]. The most common approach to address this problem is the usage of password mechanism. However, most passcodes are simple and easily guessed due to users' preference for convenience and memorability [17, 19]. Therefore, if we have some ways by which we can identify who is getting access to the system based not only on the password mechanism, we can significantly curb the threats caused by the insiders. Of various potential solutions to this problem, a promising technique is the use of keystroke-typing behavior [7, 14], which offers ability to ascertain a user's typing characteristics to verify her/his identity in a transparent manner. As a newly behavioral biometric, this behavior has been strongly driven by the need for nonintrusive verification for insider detection and monitoring applications. Compared with other biometric techniques such as fingerprints or voice, keystroke dynamics is non-intrusive, and the data is relatively effortless to collect. When a genius user getting access to the system with the interaction using a keyboard, by comparing keystroke rhythm with the genius user, insiders could be detected and rejected because their typing styles differ significantly from those of genius users. Moreover, the users' typing characteristics can be continuously analyzed during the subsequent interaction process to enforce an identity monitoring.

Typing habit differs from one to the other, so verification of people based on their typing patterns is possible, and furthermore identification is also achievable [12]. In particular, there are two tasks of interest: verification and identification. Verification checks a user's claimed identity; while identification, telling who the illegitimate person is, could fight against insider threats.

Although there are many previous research on verification and identification based on keystroke rhythm, most of the algorithms need the dataset from imposters/attackers. Identification is taken as a multi-classification and the label of every item is necessary, which limits the practical application. However, if we can build the model of every one with his/her typing data labeled and a small

Table 1: Several different studies investigating various classification algorithms and reporting evaluation results, where the diversity of conditions makes the direct comparison of the results impossible.

Source Research	Classifier	Data Collection			Ext	Feature Source			Target		Result(%)		
		Users	Task	Length		H	UD	DD	Auth	Iden	FAR	FRR	ERR
Joyce et al. (1990)[14]	Manhattan	33	Name and Password	No limit	×	√			√		6.67	0.01	-
Coltell et al. (1999)[7]	Compose	10	Password	80	×	√	√		√		0	30	-
Yu et al. (2003)[23]	SVM	21	Password	6~10	√	√	√		√		-	3.54	-
Livia et al. (2005)[2]	k-NN	30	Text	10	×	√	√	√		√	1.89	1.45	-
Hosseinzadeh et al. (2006)[11]	GMM	8	Name	10	×	√	√			√	2.1	2.4	-
Hosseinzadeh et al. (2008)[10]	GMM	41	Name	10	×	√	√	√		√	3.9	6.3	-
Villant et al. (2009)[21]	E-NN	36	Text	Hello World	×	√	√			√	-	0.5	-
Roy et al. (2011)[18]	Random Forest	28	Number	10	×	√	√	√		√	1.51	-	8.60
Bours et al. (2012)[3]	Normal distance	25	Text	Sample fixed key pairs	×	√	√			√	Number of key pair ranges from 79 to 348		
Deng et al. (2013)[8]	GMM -UBN	51	Password	10	×	√	√	√	√		-	-	5.5
Zheng et al. (2014)[24]	One-class Nearest Neighbors	53 41 42 27 25	Numeric keyboard	4/8	×	√	√	√		√	-	-	3.65 (Ave -rage on all)
Daniel et al. (2015)[4]	GMM	28	Password	8	×	√	√	√		√	-	-	30.84
	k-NN	28	Password	8	×	√	√	√		√	-	-	29.48
Margit et al. (2015)[1]	One-class k-NN	42	Password	10	√	√	√	√	√		5.0	11.0	-
	One-class Gaussian	42	Password	10	√	√	√	√	√		4.0	16.0	-

sample part of others' typing data, it will be scalable when new individuals taken into consideration without rebuilding the model of every one already in the system.

Afterward, no work yet has been done on evaluating and contrasting these classifiers, for the following reasons: 1) there is no universally accepted data set on keystroke dynamics, leading to the fact that different classifiers are tested on various dataset, and 2) there is no inconsistent evaluation measurement of the classifiers' performance. Even some of the researches are tested on the same dataset, the environment and features are various. So it is necessary to benchmark the dataset and performance indicator because only on these conditions can the indicators have reference values.

Here we develop a benchmark for user identification based on keystroke rhythm. And the remainder of this paper is organized as following: Section 2 describes the background and related work on keystroke dynamics. Section 3 introduces and visualizes the dataset. Section 4 elaborates pre-processing of the dataset. Section 5 describes the 13 algorithms. Section 6 presents the experimental results and analysis, including open set test. Section 7 gives some advice on the future work. Finally, Section 8 comes into a conclusion.

2 BACKGROUND AND RELATED WORK

In this section, a brief introduction of keystroke dynamics is offered as well as a comparison of each classifier in order to show the different methodologies of dataset and operating environment of each work.

Each person has a specific way to type his/her password, and the habit will hardly be changed once formed. Keystroke dynamics refers to a method to record one's behavioral biometric behavior data, containing the unique biometric template of typing pattern, which provides an accessible manner for individual authentication and identification [22]. Table 1 presents a brief summary of some researches using classification algorithms to identify or authenticate users by keystroke rhythm dataset. The studies above describe the experiments and results on various dataset. The first column is the reference of each study, and the others are information as following:
Classifier: the algorithm used for user identification in each study, where GMM is for Gaussian Mixture Model, and k-NN for k Nearest Neighbors.
Users: the number of users or subjects in the dataset of each study.
Task: the context that the users type when collecting dataset.
Extensibility: whether to retrain model when taking new subjects into consideration.
Feature Source: the source of features used to train and test in each study, where H stands for the latency the key is pressed, UD for keyup-keydown and DD for keydown-keydown. These abbreviations are also used in this paper.
Target: whether the study is to identify or not.
Result: FRR–the percentage of the insiders that are not detected; FAR–the percentage of the legitimate individuals that are detected as insiders; EER–the equal error rate of ROC curve.

From the researches listed in Table 1, the approaches are implemented by (1)different classifiers in the purpose whether to identify or authenticate; (2)various dataset with distinct sizes of subjects and different contexts to type; (3)different combination of typing feature; (4)different threshold to obtain different FAR/ FRR/EER. Additionally, the researches tend not to be replicated, so it's hard to make the difference clear in the same level and the results of each experiment is not comparable because it is not clear whether the feature or the dataset is the factor.

3 DATASET AND VISUALIZATION

In this section, we introduce the dataset and give a visualization of two subjects' typing rhythm.

3.1 Dataset Introduction

The dataset is from [16], Kevin S. Killourhy and Roy A. Maxion proposed the dataset in order to give a benchmark of anomaly detection [15], which only gives verification of illegitimate user accessing. It is shown that the typing rhythm may contain too many latencies for the users' thinking or finding keys if the text is not familiar to them. Therefore, the users are asked to practice the same password several times until they are comfortable with the string, and the rhythm recorded is the ones after the users have already been familiar.

In order to make the keystroke task representative, we only use UD and H time in the dataset to give a benchmark of user identification, with the following reasons: 1) the keystroke data itself has the linear additive law, which means H time plus UD is DD; 2) not all of the classifiers are capable of redundancy.

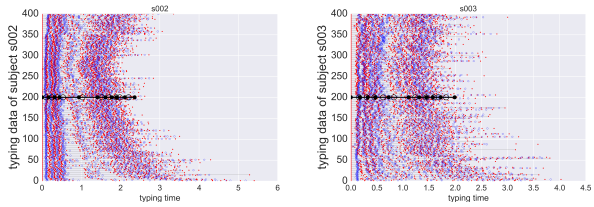

Figure 1: Typing data of subjects s002 and s003.

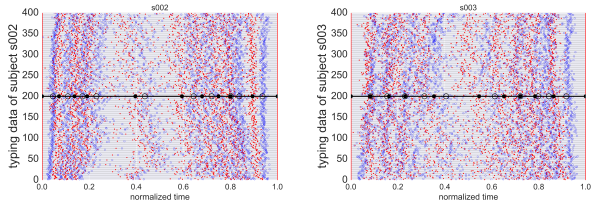

Figure 2: Normalized typing data of subjects s002 and s003.

3.2 Dataset Visualization

We display all the typing rhythm of two subjects in Fig. 1 with mean of typing data in the middle marked as large nodes, where the red nodes are for the time when key was pressed and blue for released. As we can see in the picture, the data of a certain subject is very different from mean for the various time length.

As is shown in Fig. 1, typing data for one certain subject are different in length, resulting in the unrepresentative mean. Therefore, the typing data in different length should be measured under the same metric.

4 FEATURE CONSTRUCTION

According to Section 3, we, in this section, come up with a method to pre-process the typing data. Also, by the comparison of each mean typing datum in Euclidean space, we can see the typing data appears to be in pattern after normalized.

4.1 Feature Analysis

Since data of a certain subject is in different length, a natural way to make them similar is normalization, which means normalized by the time length of each datum. Figure 2 shows the same subjects shown as in Fig. 1 after normalized. As seen in the picture, each datum looks similar to mean after normalized and it is clear to see the keystroke actions in vertical direction appear to be centralized and characterized.

From Fig. 2, we can clearly see it that some typing moments are well clustered and the mean can stand for them, while there are also moments discretely distributing. But compared with raw dataset, the typing data show the regularity and the features can be used in this way.

When comparing the mean vectors after normalization, the vectors are different from each other. If the normalized typing data of each subject is a vector in Euclidean space, it's directly to ask how they distribute in space. The Euclidean distances of every mean after being normalized is shown in Fig. 3.

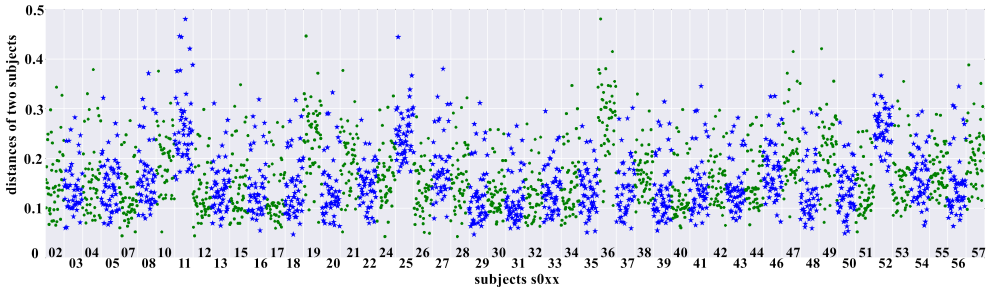

Figure 3: Euclidean distance between any two subjects in dataset after normalization.

In Fig. 3, the abscissas are the subjects in dataset. There are 51 subjects in total, thus there are 50 Euclidean distances of mean for one given subject. Each point stands for one Euclidean distance of two subjects' mean typing data. In order to show clearly, the points of one subjects are separated by green and blue. From Fig. 3, we can see that some of the typing data is far away from the others measured by Euclidean, such as s036 and s052.

4.2 Feature Extraction

The dataset samples the time click of keystroking, which is a record of typing time. When typing, even the same subject typing the same password types distinctly. Hence, the data is noised and there might be correlation between some typing latencies. Obviously, DD latency is the sum of UD and H. Consequently, we consider to extract H and UD latencies to be the feature.

5 CLASSIFIER IMPLEMENTATION

In this section, we will discuss the implementation of 13 classifiers on keystroke dynamics for identification. Besides the algorithms listed in Table 1, we also implement some other classical algorithms which can be used to identify individuals.

5.1 Classifier 1: Euclidean

Euclidean distance classification models the typing rhythm vector as a point in Euclidean space. When training, a mean vector is calculated for each subjects. And when testing, a point is classified to one class according to the distance to every mean vector of subjects. We also compare the performance of raw data with normalized ones, where normalization is to divide the data by the maximum. The classifier based on distances with mean vectors of each subject couldn't recognize the subjects not appearing in the training dataset because the classifier has not known the mean vector of the new individuals. So the classifiers based on the distance is not fit for open set test.

5.2 Classifier 2: Manhattan

This algorithm in [9] is similar to Euclidean while the distance is measured by Manhattan distance. Similar to Euclidean classifier, we implemented the Normalized Manhattan classifier and left out the open set test.

5.3 Classifier 3: Mahalanobis

This algorithm in [9] is also similar to Euclidean and Manhattan while the distance is calculated by the mean and covariance matrix of training data. All the way, the class with the minimal distance is the predicted label. Also, we implemented the Normalized Mahalanobis classifier and left out the open set test. Furthermore, Mahalanobis distance takes the covariance into consideration, leading to the fact that normalization makes no difference from raw typing data.

5.4 Classifier 4: Fisher Linear Discriminant Analysis (Fisher LDA)

Fisher Linear Discriminant is a classifier with a linear decision surface. The solution of the surface can be computed easily and Fisher LDA can also classify linear separable data in high accuracy. When training Fisher LDA model of one subject with positive label, part of the others' typing data are labeled as negative. Fisher LDA aims to find a projection direction to which after projected the positive and negative labeled data has the minimal intraclass dispersion and maximal interclass dispersion in Euclidean space.

Fisher LDA gives the decision surface after training as the model of one subject, and therefore it can test on individuals no matter whether they are in the training dataset or not. Thus, we implemented Fisher LDA on open set test. Additionally, the dataset can be normalized before training, and we test the Normalized Fisher LDA as well.

5.5 Classifier 5: Gaussian Process Classification (GPC) with RBF Kernel

A Gaussian Process Classification is a probabilistic model based on Bayesian theory and statistical learning theory, which focuses on modeling the posterior probabilities. When training, the model calculates the probability with Gaussian function of each subject. When testing, a posterior probability vector is checked to see which class the testing vector belongs to. The resulting label is the class with the minimal probability. We use RBF kernel, which can either stands for the distance of two data or is infinitely differentiable leading to a smooth classifier surface, in our experiment to calculate the similarity of two datum points. Also, GPC is tested on open set and both normalized and raw dataset.

5.6 Classifier 6-9: Support Vector Machine (SVM) with linear, polynomial, RBF, sigmoid kernel

SVM maps a vector into high dimensional feature space by a kernel function where the boundary of classification is clear to get. The decision surface can be non-linear in raw space. We use linear, polynomial, RBF and sigmoid kernel functions and show the difference among each other. We also test SVM on both normalized and raw dataset.

5.7 Classifier 10-12: k Nearest Neighbors (k-NN) based on Euclidean, Manhattan, Mahalanobis Distance

This algorithm takes the typing rhythm as the point in space. A k-nearest-neighbor algorithm is to find the k nearest points of a certain vector by Euclidean, Manhattan, Mahalanobis or any other measurement of distance. Furthermore, the resulting label is the class which appears most in k nearest neighbors. So the important parameter is the number of neighbors. If k is too large, the model would be too simple under-fitting. So it is necessary to test the classifier performance in different neighbor numbers.

From the process in k-NN algorithm, we can see it that the results of the classifier must belong to one of the labels in training dataset. Therefore, the classifier can't be applied to open set test, regardless of threshold to form new clusters, which leads to the unfairness in misclassifying new individuals. So, similar to Euclidean classifier, we leave out the open set test of k-NN.

5.8 Classifier 13: Random Forest

A random forest contains many classification trees which are single classification trees. For a certain subject, not all of the trees draw good results. However, by considering the results of all trees totally, the label given by well-performed trees will be weighted high while the bad ones low and the forest comes into an integrated result.

6 PERFORMANCE COMPARISON

In this section, we introduce the performance metrics to measure different classifiers. Then we show the results of 10-folder validation of each classifiers following the results of open set test.

6.1 Calculating Classifier Performance

For a one-against-rest algorithm, we measure the performance by false-negative identification rate (FNIR) [13] and false-positive identification rate (FPIR). In our evaluation, a false-negative occurs when the sample in class s_i is misclassified as class s_j, where $s_i \neq s_j$ for classifier of subject s_i. Similarly, a false-positive occurs when the sample in s_j misclassified as s_i, where $s_i \neq s_j$ for classifier of subject s_j. For FNIR and FPIR, we totally use F1-measure to measure the performance of classification. F1-measure is the harmonic mean of the two rates, which is closer to the minimal one. Therefore, F1-measure reflects the worse-case scenario of the classifiers.

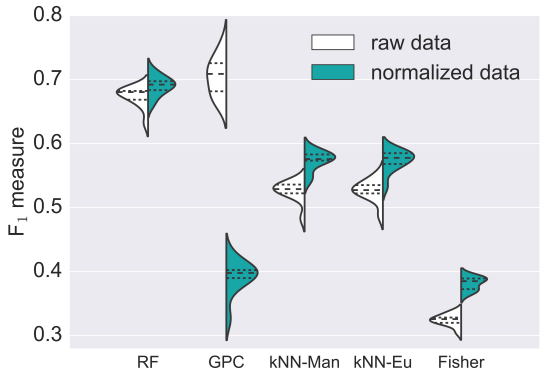

Figure 4: F1-measure of high-performance classifiers in inner testing.

6.2 Training and Testing the Classifiers

We employ the one versus rest model where the rest stands for part of other subjects, which is feasible in practice. Then we test the performance in two cases: inner and open set.

6.2.1 Inner Classification Performance. The train dataset of the classifier of one subject is formed by the typing data of the subject and all the other subjects' typing data. In 10-folder validation, a settled group is used to reduce the randomness of dataset in order to keep the classifiers tested in the same situation. F1-measure with different classifiers mentioned above is shown in Fig. 4. As for the F1-measures of some classifiers are near zero, we only display those with high F1-measure, and the whole results is shown in Table 2.

Figure 2 is plotted in violin-plot, which shows the mean and the full distribution of data. For each violin, the left side colored white is the distribution of F1-measure of classifiers on raw dataset, and the right is for normalized dataset.

For classifiers in Fig. 4, F1-measures of each classifier are high above zero, most of which get better performance after normalization. But F1-measure of Normalized GPC is not better than GPC with raw dataset. As is introduced in Section **??**, GPC models the subjects by the posterior probabilities. When normalized, the randomness of the subjects' typing data is also weakened. And GPC couldn't study the probability well. Therefore, normalization doesn't suit for GPC because it destroy the original probability distribution.

For other classifiers appearing in Fig. 4, normalization helps to improve the F1-measure for the following reasons: 1) after normalized, the typing data are all in the same length, and the typing features are more disciplinary as is shown in Fig. 2; 2) the normalized features with better dispersion makes intraclass dispersion lower and interclass dispersion higher, which suits for the classifiers, such as Fisher LDA, k-NN; 3) Random Forest uses the features directly to train classification trees, and therefore it is better to use more regular ones to classify.

There are also many classifiers not appearing in Fig. 4, such as distance based ones and SVM, and the performance is shown in Table 2. For distance based classifiers, we trained them as one-against-rest models in order to maintain the same situation with

Table 2: F1-measure of all classifiers in inner testing.

Classifier		F1 measure(%)		Improvement (%)
		Raw dataset	Norm dataset	
Distance	Euclidean	3.7255	5.6949	52.86
	Manhattan	4.8922	7.1062	45.26
	Mahalanobis	3.9423	3.9274	−0.38
Fisher LDA		32.4379	38.1318	17.55
GPC		70.4185	39.1513	−44.34
SVM	Linear	5.9230	6.0461	2.08
	Polynomial	5.8093	5.9127	1.78
	RBF	3.5625	0	-
	Sigmoid	1.9608	1.9608	0.0
k-NN	Euclidean	52.6507	57.3094	8.85
	Manhattan	52.5654	57.3178	9.04
	Mahalanobis	0.0754	0.0754	0.0
Random Forest		67.3017	69.0190	2.55
Random Guessing		1.9608	1.9608	-

other classifiers. But actually, these models are inapplicable to one-against-rest models. Taking Euclidean distance based classifier as an example, when training model of one subject, the dataset of the other subjects are considered as the negative label and the mean vector stands for them. However, the mean vector is a mixture of many other subjects' typing data, which may not have obvious clustering center. Thus the model could not classify every subjects well. The F1-measure approximates to 2%, which equals to random guessing. So the F1-measures of these classifiers are not marked in the figure.

6.2.2 Open Set Testing Performance. For a certain subject, we take the own data and part of the others' data as the training dataset. And then, the typing data of subjects not in training are used to test, which is called open set test. For classifiers such as k nearest neighbors and distance-based methods like Euclidean, they could not identify the new label not appearing in training data. Thus, we only test the following classifiers on open set test: Fisher LDA, GPC, SVM, and Random Forest.

For SVM, the F1-measures of open set test approximates to random guessing concluding that the model of every subject need to be rebuilt when a new individual arrives.

As in Fig. 5, F1-measure of Fisher LDA increases to the maximum and then declines slowly until it is stable, when training data grows. At the beginning, the training data set is not enough, and the decision surface is determined by partial data. When the training data reach to a certain range − 8 ∼ 15 individuals, the decision surface is well trained and F1-measure gets its maximum. When there are too many negative individuals in training dataset. The surface would tilt towards the positive side in order to classify correctly the negative set with large number of data. Therefore, the equalization of the training data set should be considered when using Fisher LDA in practical application.

For GPC, however, the performance isn't so well as that of inner, with F1-measure less than 40%, while F1-measure of Random Forest is up to 92% when training set grows as shown in Fig. 6. Although

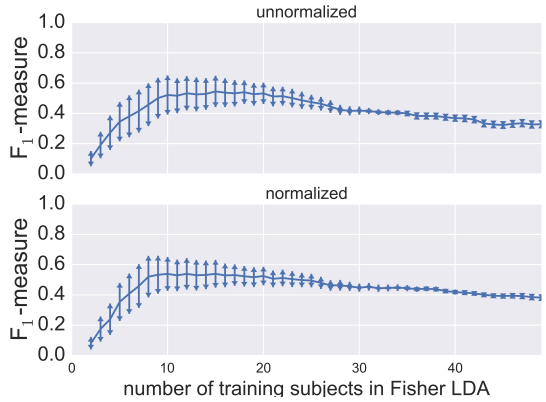

Figure 5: F1-measure of Fisher LDA in open set test.

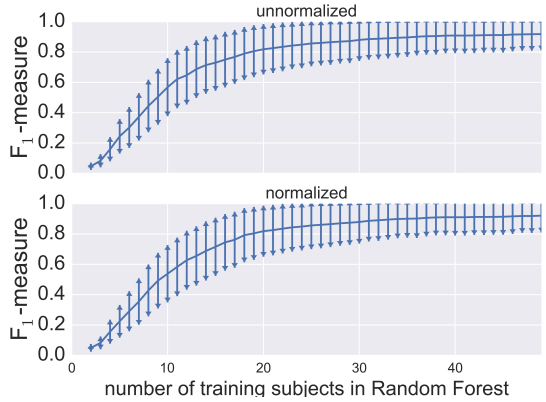

Figure 6: F1-measure of Random Forest in open set test.

GPC performs well in inner test, it only applicable to the application scenario that all the individuals are present in the training set. If a new individual arrives, the model of every existing individuals should be retrained. Meanwhile, Random Forest performs better on open set test with high F1-measure. When training set grows, F1-measure grows until it is stable. In terms of training, Random Forest is simpler than Fisher LDA, well performed on open set test because the user have not to consider the size of training dataset.

6.2.3 Number of Trees in Random Forest. For Random Forest, performing the best considering both open set and inner testing, the key parameter influencing the performance is the number of classification trees. We test the Random Forest with different number of trees. Figure 7 shows F1-measure increases when number of trees growing, and it remains unchangeable when tree number is larger than around 50, similar to the number of subjects.

For Random Forest, in order to get a better performance in actual application scenario, the number of trees should be more than the number of individuals, but the number also affects the training speed and complexity, with low speed and high complexity when the tree number is large.

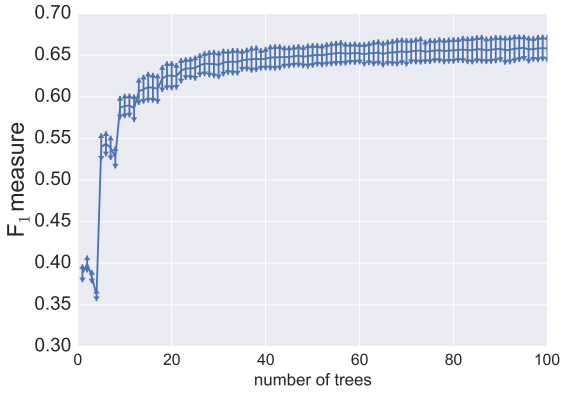

Figure 7: F1-measure varies with tree number in Random Forest.

7 DISCUSSION

Based on the researches above, we only normalized the dataset before training, but there are more methods to handle with high dimensional data such as PCA and so on. As is shown in Fig. 3, the data appear to be regularly in part of the typing moment. There are also confusion in some typing moment, and therefore, a better feature extraction method can be developed. Also, we use only the typing dataset including H and UD, while there is a need to select some better features of typing to gain F1-measure according to the physical nature of keystroking.

8 CONCLUSION

There already exist many classification algorithms proposed for identifying or authenticating individuals by the keystroke rhythm, while they are all tested in different dataset and measured by different metrics, making it's hard to compare them in the same situation. Therefore, we have tested 13 classifiers in the same dataset and environment. Meanwhile, we also came up with a method to process the keystroking dataset – normalization. We used F1-measure to measure the performance of each algorithm in both raw and normalized dataset. Combining with practical application scenarios, we test the classifiers on open set. As a result, Random Forest performed well both in inner and open set test, and can be used without rebuilding the model of existing individuals.

ACKNOWLEDGMENTS

This work was supported in part by the National Natural Science Foundation of China (61403301, 61221063), the China Postdoctoral Foundation (2014M560783, 2015T81032); the Natural Science Foundation of Shaanxi (2015JQ6216), Application Research of SuZhou (SYG201444), and Fundamental Research Funds for Central Universities (xjj2015115).

REFERENCES

[1] Margit Antal and Laszlo Zsolt Szabo. 2015. *An Evaluation of One-Class and Two-Class Classification Algorithms for Keystroke Dynamics Authentication on Mobile Devices.* Cscs. http://www.ms.sapientia.ro/~manyi/research/43.pdf.

[2] Livia C. F. Araujo, Luiz H. R. Sucupira Jr., Miguel G. Lizarraga, Lee. L. Ling, and Joao. B. T. Yabu-Uti. 2005. *User authentication through typing biometrics features.* Signal Processing, IEEE Transactions, vol. 53, no. 2.

[3] Patrick Bours. 2012. *Continuous keystroke dynamics: A different perspective towards biometric evaluation.* Information Security Technical Report. .

[4] Daniel Buschek, Alexander De Luca, and Florian Alt. 2015. *Improving Accuracy, Applicability and Usability of Keystroke Biometrics on Mobile Touchscreen Devices.* CHI. http://www.mmi.ifi.lmu.de/pubdb/publications/pub/buschek2015chi/buschek2015chi.pdf.

[5] Australian Cyber Security Centre. 2015. 2015 Cyber Security Survey. (2015). https://www.cert.gov.au/system/files/614/691/2015-ACSC-Cyber-Security-Survey-Major-Australian-Businesses.pdf.

[6] Australian Cyber Security Centre. 2015. 2015 Cyber Security Survey. (2015). https://www.acsc.gov.au/publications/ACSC_Threat_Report_2016.pdf.

[7] Oscar Coltell, Jose M. Badia, and Guillermo Torres. 1999. *Biometric identification system based on keyboard filtering.* Proc. IEEE 33rd Int. Carnahan Conf. on Security Technology, Madrid. http://www3.uji.es/~badia/pubs/carnahan99.pdf.

[8] Yunbin Deng and Yu Zhong. 2013. *Keystroke Dynamics User Authentication Based on Gaussian Mixture Model and Deep Belief Nets.* Isrn Signal Processing. .

[9] Richard O. Duda, Peter E. Hart, and David G. Stork. 2001. *Pattern classification(2nd Edition.* Wiley-Interscience. 125-134.

[10] Danoush Hosseinzadeh and Sridhar Krishnan. 2008. *Gaussian Mixture Modeling of Keystroke Patterns for Biometric Applications.* IEEE Transactions on Systems Man & Cybernetics. 38(6):816-826.

[11] Danoush Hosseinzadeh, Sridhar Krishnan, and April Khademi. 2006. *Keystroke Identification Based on Gaussian Mixture Models.* IEEE International Conference on Acoustics. 3.

[12] J. Ilonen. 2006. *Keystroke dynamics.* Keystroke dynamics. www.it.lut.fi/kurssit/03-04/010970000/seminars/Ilonen.pdf.

[13] ISO/IEC 19795-1 ISO. 2006. *Information technology - Biometric performance testing and reporting, Part 1: Principles and framework.* ISO/IEC JTC 1/SC37: Geneva. 125-134.

[14] Rick Joyce and Gopal Gupta. 1990. *Identity authentication based on keystroke latencies.* Communications of the ACM. https://www.researchgate.net/publication/220423417_Identity_Authentication_Based_on_Keystroke_Latencies?ev=pub_cit.

[15] Kevin S. Killourhy and Roy A. Maxion. 2009. *Comparing anomaly-detection algorithms for keystroke dynamics.* IEEE/ifip International Conference on Dependable Systems & Networks. http://www-cgi.cs.cmu.edu/~ksk/KillourhyMaxion2009.pdf.

[16] Kevin S. Killourhy and Roy A. Maxion. 2009. Keystroke password dataset. (2009). http://www.cs.cmu.edu/~keystroke/.

[17] Zhigong Li, Weili Han, and Wenyuan. Xu. 2014. *A large-scale empirical analysis of Chinese web passwords.* In Proceedings of the 23rd USENIX Security Symposium, San Diego, USA. http://www.usenix.org/system/files/conference/usenixsecurity14/sec14-paper-li-zhigong.pdf.

[18] Roy A. Maxion and Kevin S. Killourhy. 2010. *Keystroke biometrics with number-pad input.* IEEE/IFIP International Conference on Dependable Systems & Networks. http://www-cgi.cs.cmu.edu/~ksk/MaxionKillourhy2010.pdf.

[19] Michelle L Mazurek, Saranga Komanduri, Timothy Vidas, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor, Patrick Gage Kelley., Richard Shay, and Blase Ur. 2013. *Measuring password guessability for an entire university.* In Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security, Berlin, Germany. http://www.andrew.cmu.edu/user/nicolasc/publications/TR-CMU-CyLab-13-013.pdf.

[20] K. Mickelberg, N. Pollard, and L. Schive. 2014. *US cybercrime: Rising risks, reduced readiness Key findings from the 2014 US State of Cybercrime Survey.* US Secret Service, National Threat Assessment Center. http://web.ethisphere.com/wp-content/uploads/2014-us-state-of-cybercrime.pdf.

[21] Mary Villani, Charles Tappert, and Sung-Hyuk Cha. 2009. *Keystroke biometric identification studies on long-text input.* Doctoral dissertation, Pace University, New York. http://csis.pace.edu/~ctappert/it691-06/projects/keystroke.pdf?origin=publication_detail.

[22] Roman V. Yampolskiy and Venu Govindaraju. 2008. *Behavioural biometrics: a survey and classification.* International Journal of Biometrics. http://cubs.buffalo.edu/images/pdf/pub/Behavioral-Biometrics-A-Survey-and-Classification.pdf.

[23] Enzhe Yu and Sungzoon Cho. 2003. *GA-SVM wrapper approach for feature subset selection in keystroke dynamics identity verification.* Lecture Notes in Computer Science. http://lsia.fi.uba.ar/papers/yu03.pdf.

[24] Nan Zheng, Kun Bai, Hai Huang, and Haining Wang. 2014. *You Are How You Touch: User Verification on Smartphones via Tapping Behaviors.* IEEE, International Conference on Network Protocols. IEEE. https://www.computer.org/csdl/proceedings/icnp/2014/6204/00/6204a221-abs.html.